



Dynamic Hybrid (Cloud) Applications

Introducing the INAETICS project



Bram de Kruijff
Luminis Technologies



INtelligent robust ArchitecturE for TIme Critical Systems

An **open collaboration effort** that aims to define and demonstrate a **dynamic service oriented reference architecture** that addresses the requirements of **time critical systems** in a **broad range of domains** by providing a **single design and implementation space** for all subsystems, **irrespective of control strategy**

Open innovation



Define an open reference architecture based on a shared vision that can be applied freely in a broad range of domains.

- Implementations in Open Source
- Spearheaded and funded by...

THALES

UNIVERSITY OF TWENTE.



alliander



investing
in **your** future
European Regional Development Fund
European Union

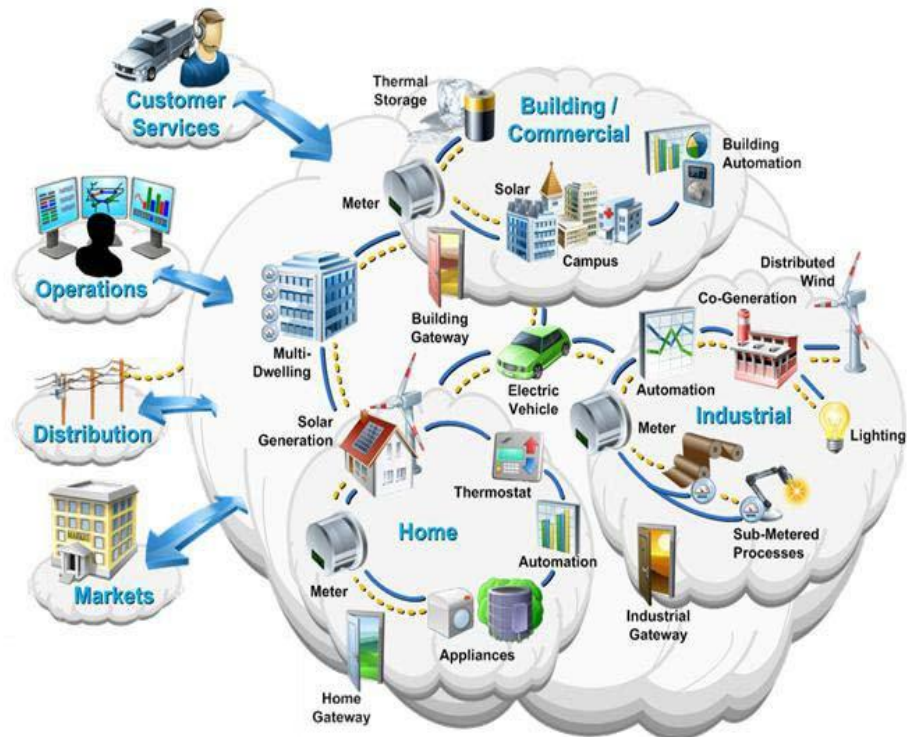


Time Critical systems



(Ultra) Large Online Systems that must reliably perform mission-critical functions, bridging operational and informational domains.

- Multiple Control strategies
- Functional Adaptive behaviour
- Technologically heterogeneous
- Geographically dispersed





Consistent design space

Define a unified design (and implementation) space across functional and technological domains that supports runtime evolution.

- Architectural consistency in systems design
- No architectural impedance between domains
- Runtime system evolution of deployments
- Adaptive system behaviour to context changes



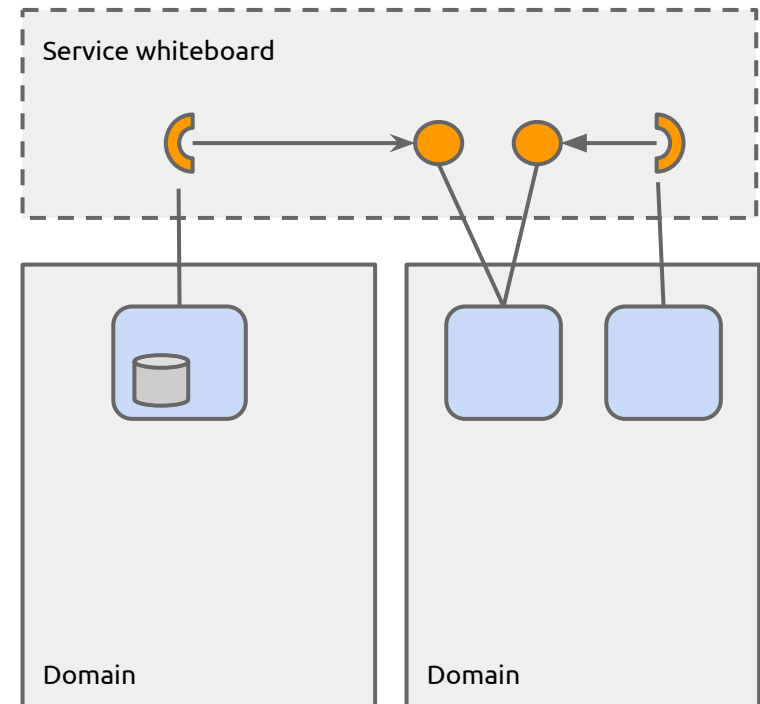
INAETICS

Dynamic Services Architecture



Dynamic Services Architecture has the properties to address the requirements for Time Critical System domains.

- Consistent design space
 - Published contracts
 - Service lifecycle
 - Multiple paradigms
- Runtime system evolution
 - Semantic versioning
 - Modular implementation
 - Data stewardship
 - (Native) OSGi

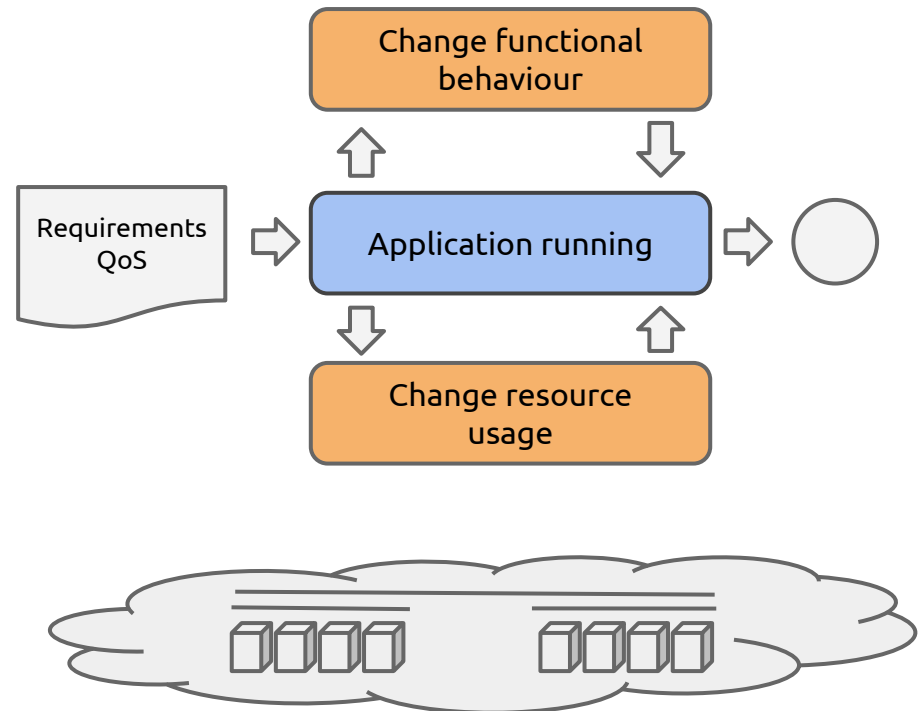


Continuous Deployment



Application deployment is an autonomous and continuous process that optimizes the application's performance given the capabilities of the available resources and budget.

- Managed Requirements
 - (Non-) Functional / QoS
 - Budget
- Resource Capabilities
 - (Non-) Functional / QoS
 - Cost
- Runtime metrics
 - Resource utilization
 - QoS reporting

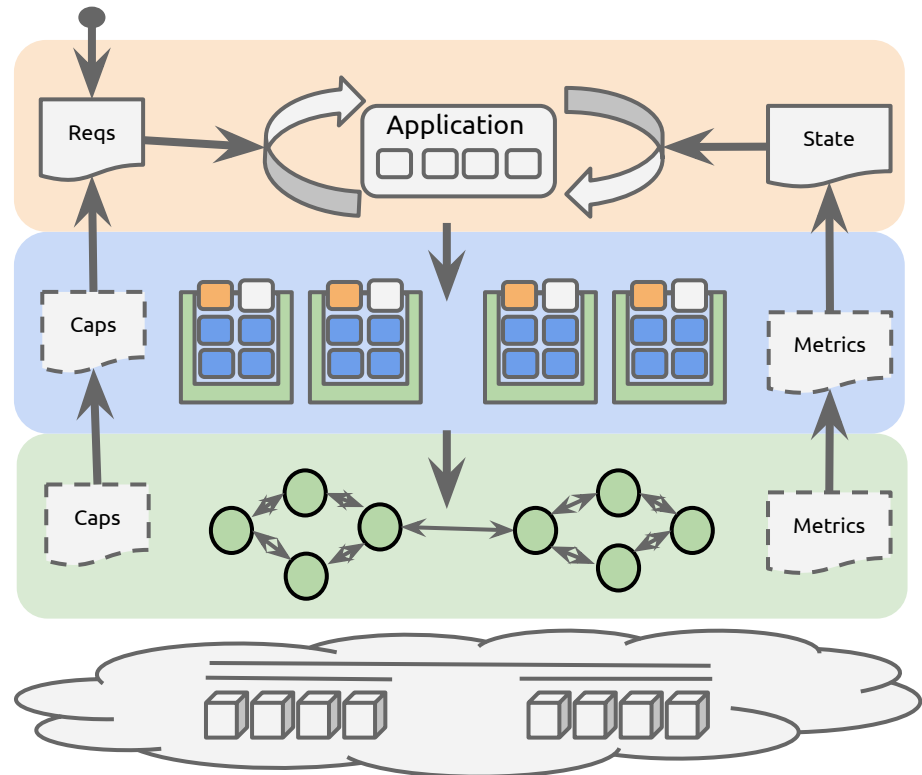


Architectural layering



The INAETICS architecture acknowledges a functional service layering with distinct responsibilities to support architectural and system evolution.

- (Application services)
- Coordination services
- Container services
- Fabric services
- (Network/Resources)

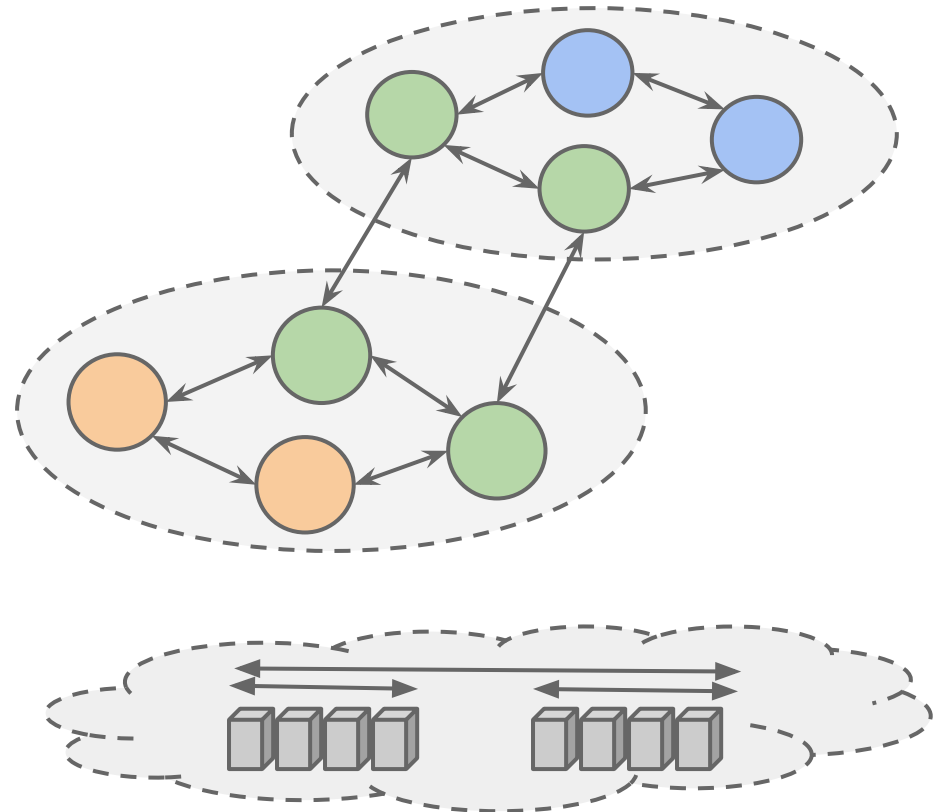




Fabric layer

The fabric layer is an extensible resource and network abstraction layer. It provides the basic infrastructure for the system state and interaction.

- Peer-to-peer network
 - Nodes/Zones
 - Zone coordination
 - Messaging/Relays
- Node capabilities
 - Non-functional / QoS
 - Functional endpoints
 - Component containers

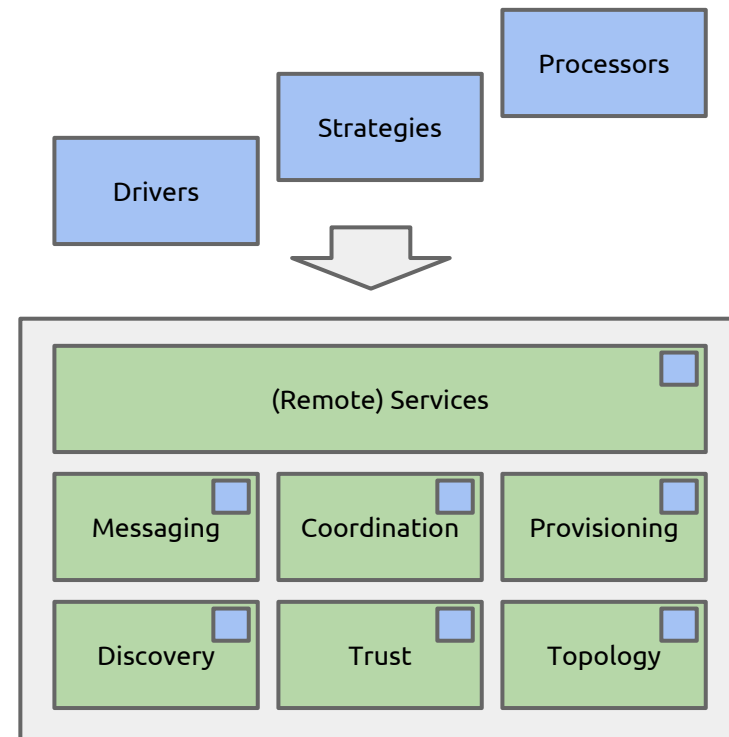




Node agents

The fabric is implemented by node agents that provide the core infrastructural services and base containers.

- Polyglot implementations
 - OSGi (Amdatu Platform)
 - Native OSGi (Apache Celix*)
 - Custom/Embedded
- Core services
 - Discovery/Trust
 - Topology/Messaging
 - Role/Task coordination
 - Initial provisioning
 - Remotes Services



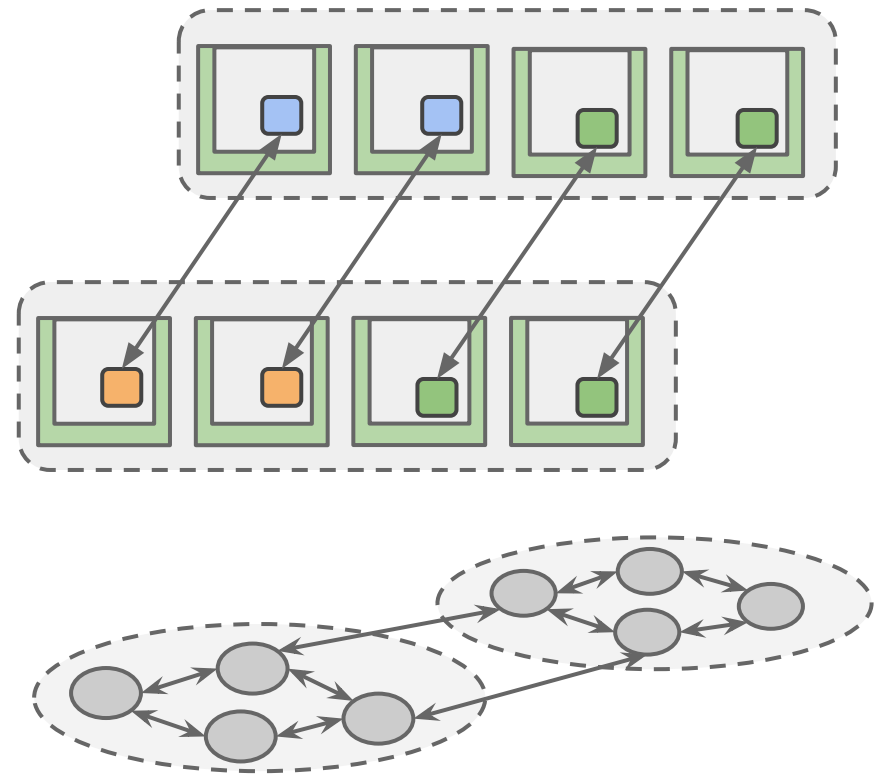
*) Apache Celix is an ASF incubator project



Container layer

The container layer is an extensible application component layer. It provides life-cycle and context to application components.

- Component runtime
 - Provisioning
 - Application context
 - Supporting services
 - QoS monitoring
- Container capabilities
 - Non-functional / QoS
 - Functional endpoints
 - Multi-tenancy

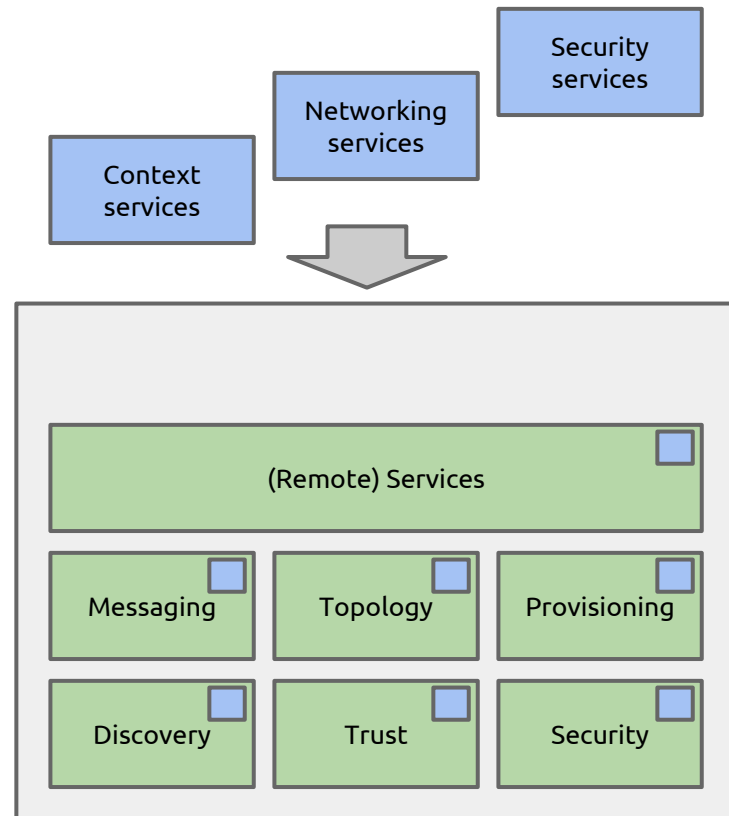




Container services

The container services are dynamic services provisioned onto node agents to support local application components with management, context and supporting services

- Provisioning
 - DeploymentAdmin
 - Apache ACE
- Application Context
 - Service visibility
 - Amdatu Platform
- Dynamic services
 - Networking services
 - Security services

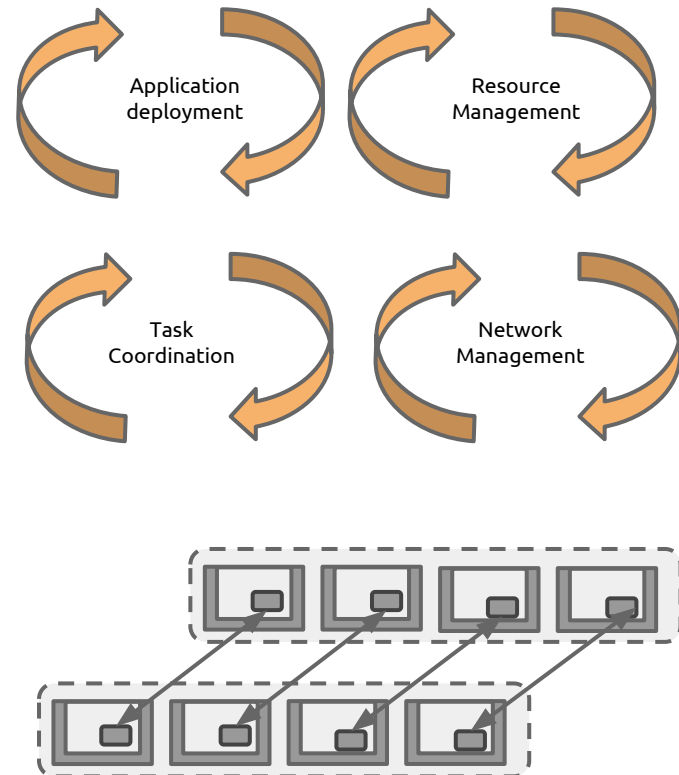




Coordination layer

The coordination layer is an extensible application management layer. It provides deployment and monitoring of the system and applications.

- System management
 - Application deployment
 - Resource management
 - Network management
 - Security management
- Application support
 - Role/Task coordination
 - Priority based scheduling
 - Load balancing

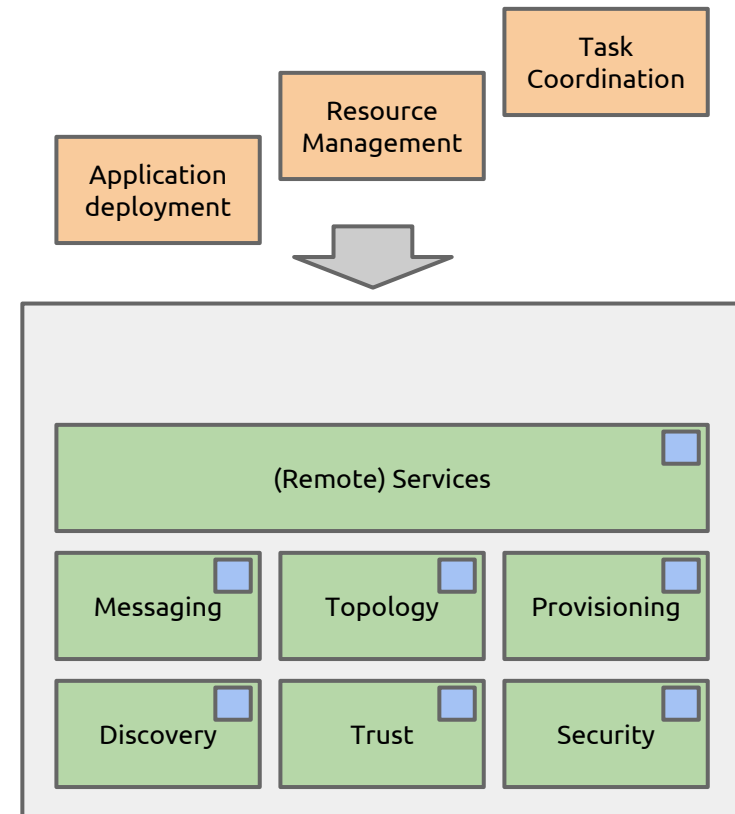




Coordination services

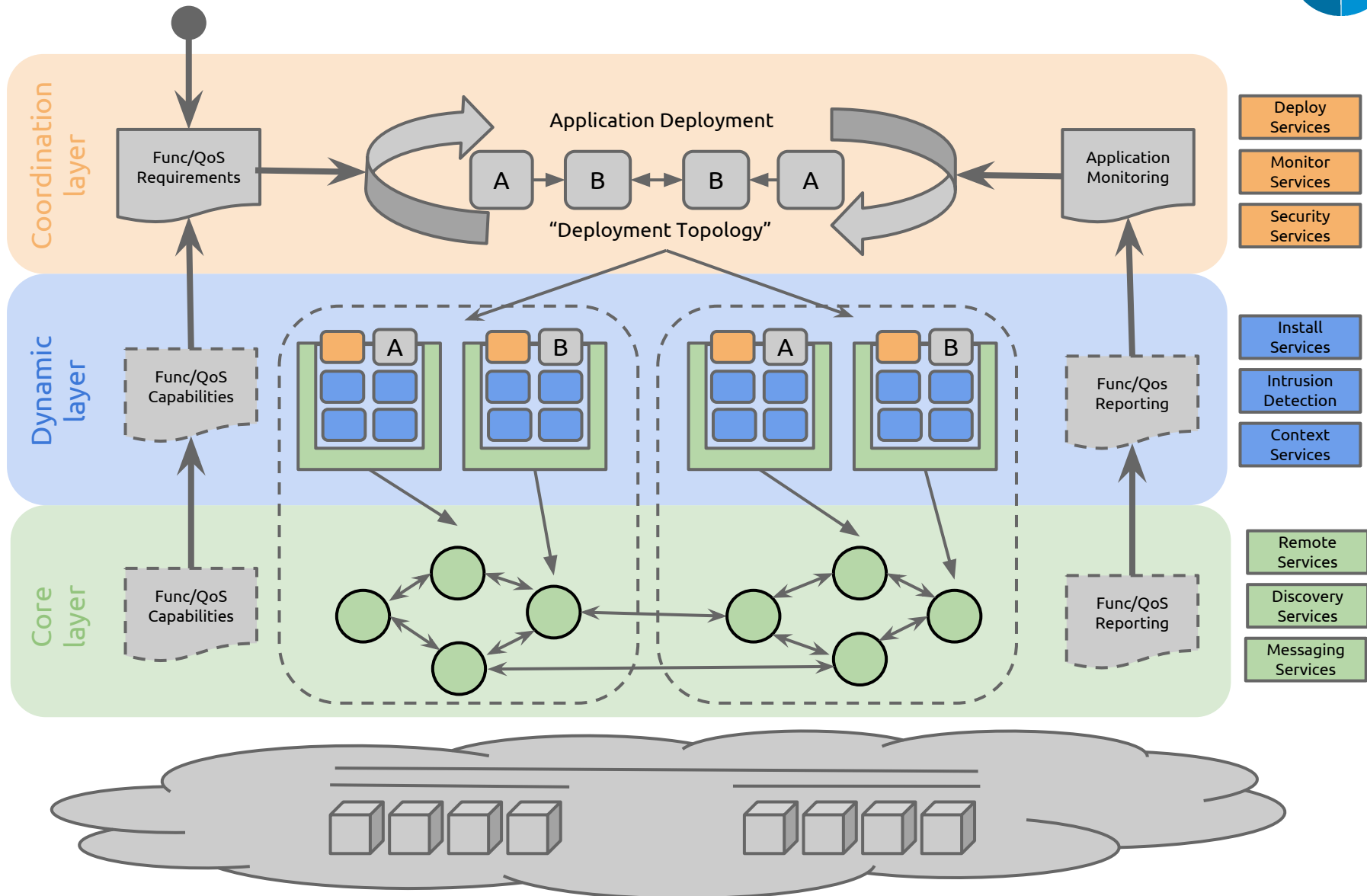
The coordination services are selectively provisioned onto node agents in the system to provide redundant, optimized and fail-safe coordination facilities

- Application deployment
 - Requirement/Capabilities
 - QoS Monitoring
- Resource Management
 - Apache JClouds*
 - OpenStack
 - OpenContrail



*) Apache JClouds is an ASF incubator project

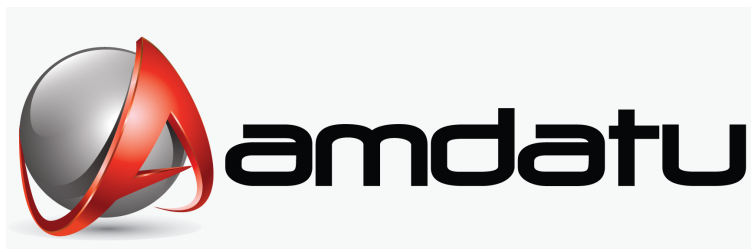
Conceptual overview



INAETICS is ...



- A dynamic services architecture that provides a single consistent design space across domain and technology boundaries
- A fully dynamic deployment, resource and network management model allowing QoS driven runtime adaptive behaviour of systems
- A runtime platform being implemented in OSGi and Native OSGi/C by the Amdatu Platform and Apache Celix Open Source projects
- ... under construction ;)



Thank you!



Continuous Automated Deployment with Apache ACE
Thursday at 11:15 - 11:50
Schubartsaal

contact@inaetics.org
<http://www.inaetics.org>

bram.dekruijff@luminis.eu
<http://www.luminis-technologies.com>

